Midterm 2
APPM/MATH 4650 Fall '20 Numerical Analysis

Due date: Saturday, October 31, before 3 PM, via Gradescope and Canvas. **Instructor**: Prof. Becker

Instructions There are **two components** to this midterm, with separate rules:

50 points The online Canvas "quiz" which is true/false or multiple choice. You have up to **45 minutes** to complete this. Please do this *before* doing the second component of the test. This component of the test is **closed note, closed book, closed computer/calculator/phone**, meaning that you should not use any resource other than your mind and scratch paper.

100 points The written part (with questions listed below on this document). You have up to **2 hours** to complete this. This component of the test is **open note**, and you **can use the Burden and Faires textbook** (9th or 10th edition), and you can **use Matlab/python or a calculator for simple things** (i.e., for programming things from scratch, and using low-level functionality like core Numpy routines). You can also use the instructor's notes on github, your own notes, the instructors homework solutions, and your own solutions, but do *not* use the class demos. You may *not* use Matlab/python's builtin root-finders, for example. You **cannot use the internet** other than for uploading to Gradescope, or checking Canvas/Piazza, or connecting to `colab` or something similar. In particular, you may **not use wikipedia or stackexchange websites**.

This exam only works if you follow the CU Honor Code. Violating the rules of the exam are simply not fair to your fellow students. Do not discuss any aspect of this exam with other students until after 3 PM Saturday.

Have questions? Please ask on Piazza (and use your judgment about whether to make it a private or public post). Prof. Becker will be actively answering questions from 3–5 PM Friday. After that, he will check for questions infrequently, though TAs may occasionally check too.

On neither portion of the exam are you allowed to use a symbolic math program (graphing calculator, Mathematica, Maple, Desmos, Sage, Wolfram Alpha, Matlab/Python with symbolic packages, etc.). You *can* use a calculator if you want. You *can* write your answers on a tablet if you like (alternatively, write on paper and take a picture or scan it).

Problem 1: **[30 points] Interpolation** Suppose we have 3 distinct nodes $\{x_0, x_1, x_2\}$ and we have the following information about a function $f \in C^2(\mathbb{R})$:

x_i	$f(x_i)$	$f'(x_i)$	$f''(x_i)$
x_0	A	D	G
x_1	B	E	H
x_2	C	F	I

For each of the variants of interpolation below, answer these **two** questions: (1) what is/are the degree of the polynomial(s) involved, and (2) which of the data $\{A, B, \ldots, H, I\}$ would you need to find the interpolant?

a) Polynomial interpolation

b) Piecewise polynomial interpolation

c) Cubic splines with natural boundary conditions

d) Hermite interpolation

Problem 2: [25 points] **Richardson Extrapolation** Suppose we are given some quadrature rule and use it to approximate an integral. The column "c_1" in the table below shows the estimates from this quadrature rule for different values of n (where the node spacing h is $O(1/n)$). The column "c_2" of the table shows the first column of Richardson extrapolation applied to this quadrature rule. Your job is to continue the Richardson extrapolation to supply the **first relevant entry in the "c_3" column, to 15 decimal places**, and **show your work and explain your reasoning**. You do not need to give the values for the rest of that column, only the first entry.

Hints: You may want to plot the data in order to estimate the convergence rates (you can use Matlab or Python to plot); you do not have to turn in your plot, though it may help you get partial credit in case you get the wrong answer. To estimate the error, a reasonable idea is to use the last entry in the "first column of Richardson" column, as it should be the most accurate.

If you want to plot the data, you don't have to type it in; you can download each column at:

- https://github.com/stephenbeckr/numerical-analysis-class/blob/master/Exams/column_c0.txt
- https://github.com/stephenbeckr/numerical-analysis-class/blob/master/Exams/column_c1.txt
- https://github.com/stephenbeckr/numerical-analysis-class/blob/master/Exams/column_c2.txt

c_0 n	c_1 quadrature rule	c_2 first column of Richardson	c_3 second column of Richardson
8	3.141853320742374		
16	3.141634186673505	3.141602881806524	
32	3.141598567141290	3.141593478636688	?
64	3.141593444476827	3.141592712667618	
128	3.141592755917011	3.141592657551323	
256	3.141592666605240	3.141592653846416	
512	3.141592655231017	3.141592653606128	
1024	3.141592653795842	3.141592653590817	
2048	3.141592653615609	3.141592653589862	
4096	3.141592653593017	3.141592653589790	

Problem 3: [25 points] **IVP** Consider the following ODE initial value problem

$$\frac{d}{dt}y(t) = 10y^2, \quad y(0) = 0.1$$

We're interested in solving this for the time interval $[0, 1]$.

a) Does this ODE satisfy the relevant Lipschitz continuity condition? Explain and/or show your work

b) Can we guarantee existence and uniqueness of a solution to this IVP on $[0, 1]$? Explain why or why not

c) Do we have a "global" error bound for Euler's method applied to this IVP? That is, is there an error bound that applies to all the time points $\{0, h, 2h, \ldots\}$ within our interval $[0, 1]$? Explain why or why not

d) Write out the approximation of the solution $y(t)$ using Euler's method with a stepsize of $h = 1/4$ for time $[0, 1]$. You may use Matlab/Python/calculator etc. to do the computation. Your answer should be a two-column table with t and the approximation to $y(t)$, with all numbers showing at least 3 decimal places. *Note*: If you use Matlab/python, you do not need to include your code, but if you do, it will help you get partial credit in case you get the wrong answers. If you use a calculator or do it by hand, you may want to include your scratch work to help get partial credit.

Problem 4 is on the next page

Problem 4: [**20 points**] **Quadrature** Consider the following quadrature rule

$$Q = \frac{b-a}{18}\left(-3f(x_0) + 16f(x_1) + 5f(x_2)\right), \quad \text{where} \quad x_0 = a,\ x_1 = a + \frac{b-a}{4},\ x_2 = b$$

for approximating $I = \int_a^b f(x)\,dx$. This quadrature rule was created by finding the unique polynomial interpolant (of the relevant degree) on the nodes $\{x_0, x_1, x_2\}$ and then integrating this polynomial. In other words, we constructed the polynomial $p(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + f(x_2)L_2(x)$ using the Lagrange polynomials L_i, and then $Q = \int_a^b p(x)\,dx$.

a) Assuming $f \in C^\infty[a,b]$ and that $|f'(x)| \le M_1$, $|f''(x)| \le M_2$, $|f'''(x)| \le M_3$, and $|f^{(4)}(x)| \le M_4$ for all $x \in [a,b]$, derive an expression for the error $|I - Q|$ in terms of only a, b and the M_i.

   Hint:
$$\int_0^h \left| x\left(x - \frac{h}{4}\right)\left(x - h\right)\right| dx = \frac{71}{1536}h^4. \tag{1}$$

b) What is the order of exactness (aka order of accuracy) for this quadrature rule? Explain what this means, and how you came to this conclusion.